Es handelt sich bei dieser Arbeit um eine exemplarische Studie. Die eigenen Untersuchungen zum Domain Shift werden anhand eines Klassifikationsbeispiels im Bereich der Bilderkennung untersucht. Es ist wichtig zu erwähnen, dass das Experiment nicht aussagen solle, dass ein Domain Shift immer durch einen bestimmten Vorverarbeitungsschritt reduziert werden kann. Vielmehr möchte getestet werden, ob im Falle der emotionalen Klassifikation von gezeichneten Gesichtern eine Verbesserung durch die - in der Implementierung benannte - Cartoonify-Funktion, die unten im Detail behandelt wird, erzielt werden kann. Der Domain Shift entsteht dadurch, dass für das Training des Klassifikators Gesichter von realen Menschen verwendet werden, während das Ziel ist, gezeichnete erkennen zu können. Des Weiteren wird beispielhaft getestet, wie sich leichte Veränderungen, wie das Anheben der Augenbrauen, auf die Prognose auswirkt.

Im Laufe der Arbeit werde ich mich die hauptsächlich die englischen Fachbegriffe verwenden, da diese auch im deutschsprachigen Raum verwendet werden bzw. es keine gute Übersetzung gibt.

Bei der Implementierung handelt es sich grundsätzlich um eine Transfer Learning Aufgabe. Transfer Learning wird gerne im Bereich des Machine Learnings verwendet, da Ressourcen, im Sinne von Rechenkapazitäten, durch die Wiederverwendung bereits trainierter Modelle gespart werden können. In den meisten Fällen wurden diese Modelle auf sehr großen Datensätzen trainiert, was für eine größere Varianz in den Daten sorgt. Dies größere Varianz erzielt einen regularisierenden Effekt. Bei Convolutional Neural Networks, welche im Bereich der Computer Vision stark vertreten sind, spiegelt sich das z.B. durch eine hohe Anzahl an unterschiedlichen Filterkernels wider, dies es ermöglichen verschiedenste Formen in Bilder zu erkennen.\\

Nennenswert für diese Arbeit ist das auf ImageNet vortrainierte ResNet-18 Modell, welches für die Implementierung verwendet wird. Somit können die bereits genannten Vorteile genutzt werden, sowie sich auf den Domain Shift fokussiert werden.\\

Ein wesentlicher Nachteil des Transfer Learning ist der Fehler, der durch den bereits genannten Domain Shift entstehen kann. Grund ist, dass Modelle auf Daten trainiert werden, die nicht der Verteilung der Daten, die ein Modell sieht, wenn es eingesetzt wird, entspricht. Mit einer großen Varianz an Daten versuch man dem entgegenzuwirken. Oft sind diese Mengen aber nicht vorhanden.\\

Deutlich wird es in dieser Arbeit durch die Verwendung von echten und gezeichneten Bildern. Die Aufgabe des Modells bleibt gleich, es ändern sich nur die Daten/Bilder, die klassifiziert werden müssen, bzw. deren Verteilung.

Die Motivation hinter dieser Arbeit entstand durch eine Recherche zur emotionalen Klassifikation von Cartoons/Anime Gesichtern. Es ließen sich keine öffentlichen Datensätze finden, die geeignet für die Aufgabe wären. Jedoch fand man Daten für dieselbe Klassifikationsaufgabe, nur anstatt Cartoons/Anime mit menschlichen Gesichtern. Gefundenen Arbeiten zu dem Thema beschäftigten sich meisten mit der Erkennung von Gesichtern, aber nicht mit der Klassifizierung, oder verfolgten ein anderes Ziel. Interessant ist es auch wegen der Merkmale, die im gezeichneten Medium verwendet werden, um bestimmte Emotionen darzustellen, wie z.B. Dampfwolken, die aus den Ohren strömen, welche mit Wut assoziiert werden. Für ein Modell, welches auf menschlichen Bildern trainiert wird, kann das für Rauschen sorgen, da diese Merkmale eben nicht im Training vorhanden sind. Von den verschiedensten Zeichenstilen noch gar nicht zu sprechen. Deswegen wäre auf lange Sicht das Ziel ein eigenen Datensatz für dieses Problem zu erstellen, wo mir das unten implementierte Modell ggf. helfen könnte.

Die Implementierung kann in dem gelisteten GitHub Repository gefunden werden \cite{}. Am besten klont man es und für die Jupyter Notebook-Datei „FaceRecog\_clean.ipynb“ aus, um der Beschreibung der Implementierung besser folgen zu können.

Die Daten, die in für das Experiment verwendet werden, sind öffentlich auf Kaggle zu finden \cite{kaggleData}.\\

Der Datensatz besteht aus acht Ordner. Der Name des Ordners bestimmt das Label der Bilder, die sich in dem jeweiligen Ordner befinden. Somit liegen Bilder zu den Emotionen "anger", "contempt", "disgust", "fear", "happiness", "neutrality", "sadness" und "surprise" vor. Die Auflösung beträgt über alle hinweg 224 mal 224 Pixel. Des Weiteren liegen die Daten als Graubilder vor \ref{fig:myData}. Das heißt u.a.: Sie haben, wenn sie als Tensor eingelesen werden, eine Dimensionalität von [224 x 244] besitzen. ResNet erwartet für das Training einen Input der Form [Batchsize x Farbkanäle x Breite x Höhe]. Deswegen werden die Farbkanäle künstlich durch die „\_add\_RGBLayers“-Funktion ergänzt.\\

Bei Betrachtung der Klassenverteilung fällt auf, dass besonders die Klasse ‘contempt‘ unterbesetzt ist, während in ‘happiness‘ mit Abstand die meisten Bilder vorhanden sind. Diese Imbalance muss später beim Training berücksichtigt werden, damit das Modell nicht einfach die a-priori Verteilung auswendig lernt.

Die Bilder, auf denen getestet wird, ob der Domain Shift verringert werden konnte, sind alle selbst erstellt \ref{fig:myTest}. So ist es einfacher simple Änderungen am Bilde selbst durchzuführen. Entsprechend der Anzahl an Emotionen wurde für jede Klasse ein Bild erstellt, mit der Intension, dass sich die Gesichter zueinander unterscheiden. Beim Einlesen der Bilder muss darauf geachtet werden, dass sie ursprünglich im Format [W x H x 4] vorliegen, wobei die Höhe und Breite für jedes Bild individuell sind. Die vierte Ausprägung in der dritten Dimension gibt den Transparenzwert des jeweiligen Pixels an. Dies kann ignoriert werden und wird in der Implementation abgeschnitten.

Im Wesentlichen wird auf zwei Aspekte detaillierter eingegangen: Das Dataset und das verwendete Modellarchitektur. \\

Das Dataset ist so aufgebaut, dass es zur Initialisierung nur den Pfad zum Oberordner und eine Liste, die die zu verwendeten Emotionen beinhaltet, benötigt. Die Liste muss dabei dieselben Namen aufweisen, wie die Ordner, die die Bilder enthalten. Der Pfad und die Liste spiegeln somit die Ordnerstruktur des Datensatzes wider. Über die Liste kann außerdem gesteuert werden, welche Bilder einer Klasse man in das Dataset mitaufnehmen möchte. So könnte man das Acht-Klassenproblem in mehrere binäre Klassifikationsproblem aufteilen.\\

Der nächste String der übergeben werden kann, bestimmt die Art des Filters, der auf die Bilder angewandt werden soll oder ob kein Filter angewandt wird. Zum Stand der Abgabe ist nur ein Cartoonify-Filter implementiert. Das Dataset kann aber in für fortlaufenden Projekte um andere Filter, die in Frage kommen würden, erweitert werden.\\

Um das Problem der unbalancierten Daten anzugehen, bietet das Dataset einen eigenen Lösungsansatz, der mit dem Parameter \textit{balanced\\_sampling} kontrolliert werden kann. Wenn dieser Parameter auf \textit{True} gesetzt ist, werden gleichmäßig Samples aus allen Klassen gezogen. Die Klassen, die relative unterbesetzt sind, werden dabei automatisch Upsampled und die Klassen, die relative viele Bilder besitzen, werden Downsampled. \\

Die Größe der Bilder kann außerdem festgelegt werden und ob sie als Graubilder, also ohne Farbkanäle, zurückgegeben werden sollen oder ein ggf. künstliche die Farbkanäle hinzugefügt werden sollen. Der Gedanke dahinter war, dass vielleicht unterschiedliche Auflösungen zu unterschiedlich guten Ergebnissen beim Training führen könnten.\\

Abschließend ist der letzte Parameter zuständig für die Augmentierungen, die an das Dataset übergeben werden können. \\

Als Modell wird der Stamm einen vortrainierten ResNet-18 verwendet. Um es an den Anwendungsfall anzupassen ist der Kopf, als der letzte Layer im Netz, abgeschnitten und durch einen eigenen ersetzt. Man kann hierfür bei der Initialisierung des Netzes auch bestimmen, wie vielen Klassen unterschieden werden sollen. Der ResNet-18 Stamm ist eingefroren, um beim Training Ressourcen zu sparen und Overfitting zu vermeiden. Somit werden beim Training nur die Layer des Kopfs angepasst und ResNet-18 fungiert als Merkmals-Extraktor.